

Ansible Rest APIモジュールによるCIMCの設定

内容

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[CIMC APIの概要](#)

[コンフィギュレーション](#)

[1. CIMC管理対象オブジェクト\(MO\)のクラスまたはDNを見つけます](#)

[1a. APIを使用してCIMCにログインし、Cookie情報を取得します](#)

[1b. APIクエリメソッドconfigResolveDnを使用してすべての管理オブジェクト\(MO\)情報を取得する](#)

[例1: タイムゾーンのクラスとDNを照会する](#)

[例2: ホスト名のクラスとDNを照会する](#)

[2. REST APIによるCIMCの管理](#)

[APIメソッドconfigResolveClassを使用して情報を取得する](#)

[APIメソッドconfigConfMoを使用して設定を変更する](#)

[3. CIMCオートメーション設定ワークフローの例](#)

[関連情報](#)

はじめに

このドキュメントでは、Ansible REST APIモジュールを使用してCisco Integrated Management Controller(CIMC)を設定する方法について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- UCS CIMC
- API
- アンゼブル

使用するコンポーネント

- UCS C220-M4、バージョン4.1(2f)
- postmanおよびansibleバージョン2.14.5を実行しているクライアント

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、クリアな(デフォルト)設定で作業を開始して

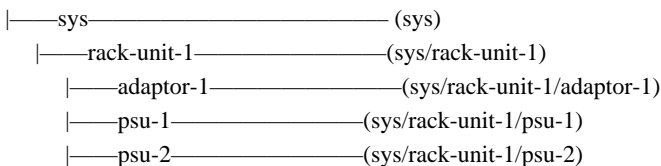
います。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

CIMC APIの概要

Cisco UCSを構成するすべての物理コンポーネントと論理コンポーネントは、MITとも呼ばれる Hierarchical Management Information Model(MIM)で表されます。ツリー内の各ノードは、管理オブジェクト(MO)、またはその管理状態と操作状態を含むオブジェクトのグループを表します。

階層構造は最上位(sys)から始まり、親ノードと子ノードが含まれます。このツリーの各ノードは管理対象オブジェクトであり、Cisco UCSの各オブジェクトには、ツリー内でのオブジェクトとその場所を示す一意の識別名(DN)が割り当てられています。管理対象オブジェクトは、CPU、DIMM、アダプタカード、ファン、電源装置などのCisco UCSリソースを抽象化したものです。

CIMC MIM構造の図：



オブジェクトの名前：

- DN：識別名を使用すると、ターゲット・オブジェクトを明確に識別できます。
- RN：相対名は、親オブジェクトのコンテキスト内のオブジェクトを識別します。

たとえば、次の識別名を使用します。

```
<dn = "sys/rack-unit-1/adaptor-1/host-eth-eth2"/>
```

4つの相対名で構成：

```
topSystem MO: rn="sys"
computeRackUnit MO: rn = "rack-unit-1"
adaptorUnit MO: rn="adaptor-"
adaptorHostEthIf MO: rn="host-eth-"
```

この記事で使用されるAPI:

- 認証：aaaLogin ログインの初期メソッド。有効なcookieを取得するには、aaaLoginメソッド

ドを使用します。

- クエリー: configResolveDnです。DNでオブジェクトを取得します。
- 設定 : configConfMo configConfMoメソッドは、管理対象オブジェクト(MO)の1つ以上のプロパティを設定するために使用されます。設定されるMOは、識別名(DN)によって一意に識別されます。



注 :

多くのクエリメソッドには、ブール値 (true/yesまたはfalse/no) を受け入れる inHierarchical引数が含まれています。trueに設定すると、メソッドは階層内のすべての子オブジェクトを返します。

コンフィギュレーション

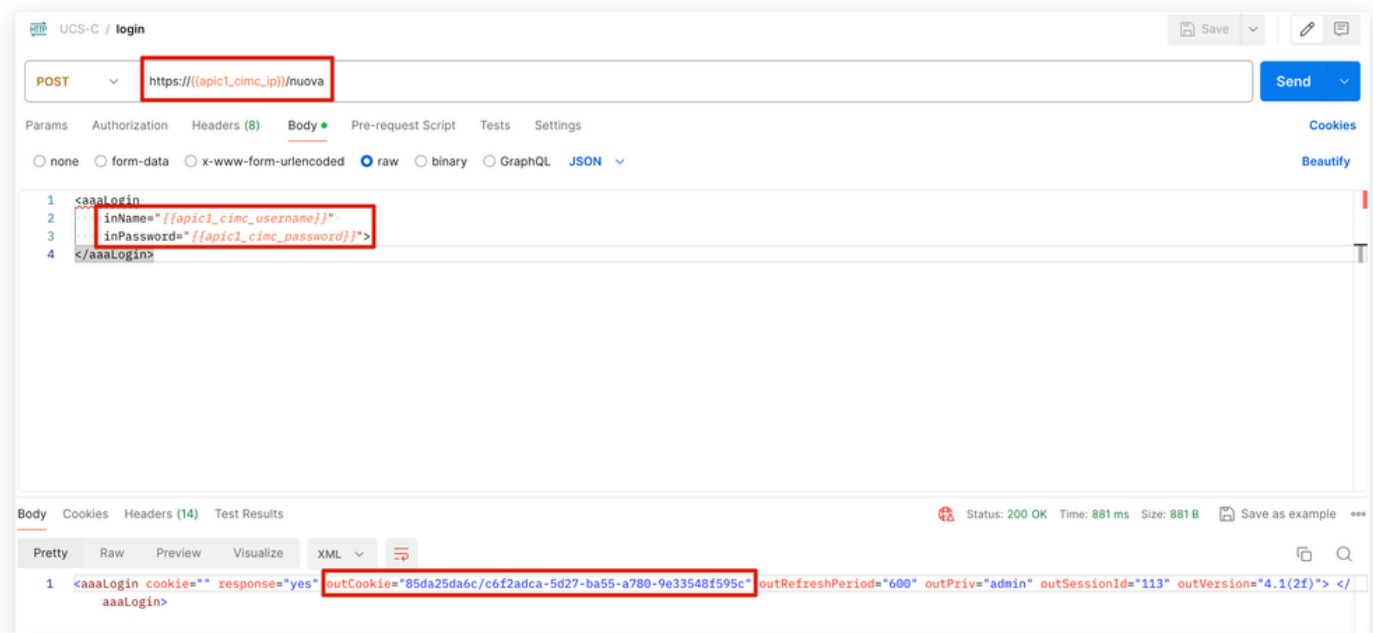
1. CIMC管理対象オブジェクト(MO)のクラスまたはDNを見つけます

APIを介してCIMCの設定を自動化するには、設定する管理対象オブジェクト(MO)に関連付けられた特定のクラスまたは識別名(DN)情報を決定することが不可欠です。

1a.APIを使用してCIMCにログインし、Cookie情報を取得します

POST要求をhttps://{{apic_cimc_ip}}/nuova に送信し、aaaLoginメソッドを指定します。ユーザ名とパスワードを入力します。

API応答からcookieをコピーします。



または、curlを使用してcookie情報を取得することもできます。

```
curl -k -d "
```

```
" https://apic_cimc_ip/nuova
```

1b.APIクエリメソッドconfigResolveDnを使用してすべての管理オブジェクト(MO)情報を取得する

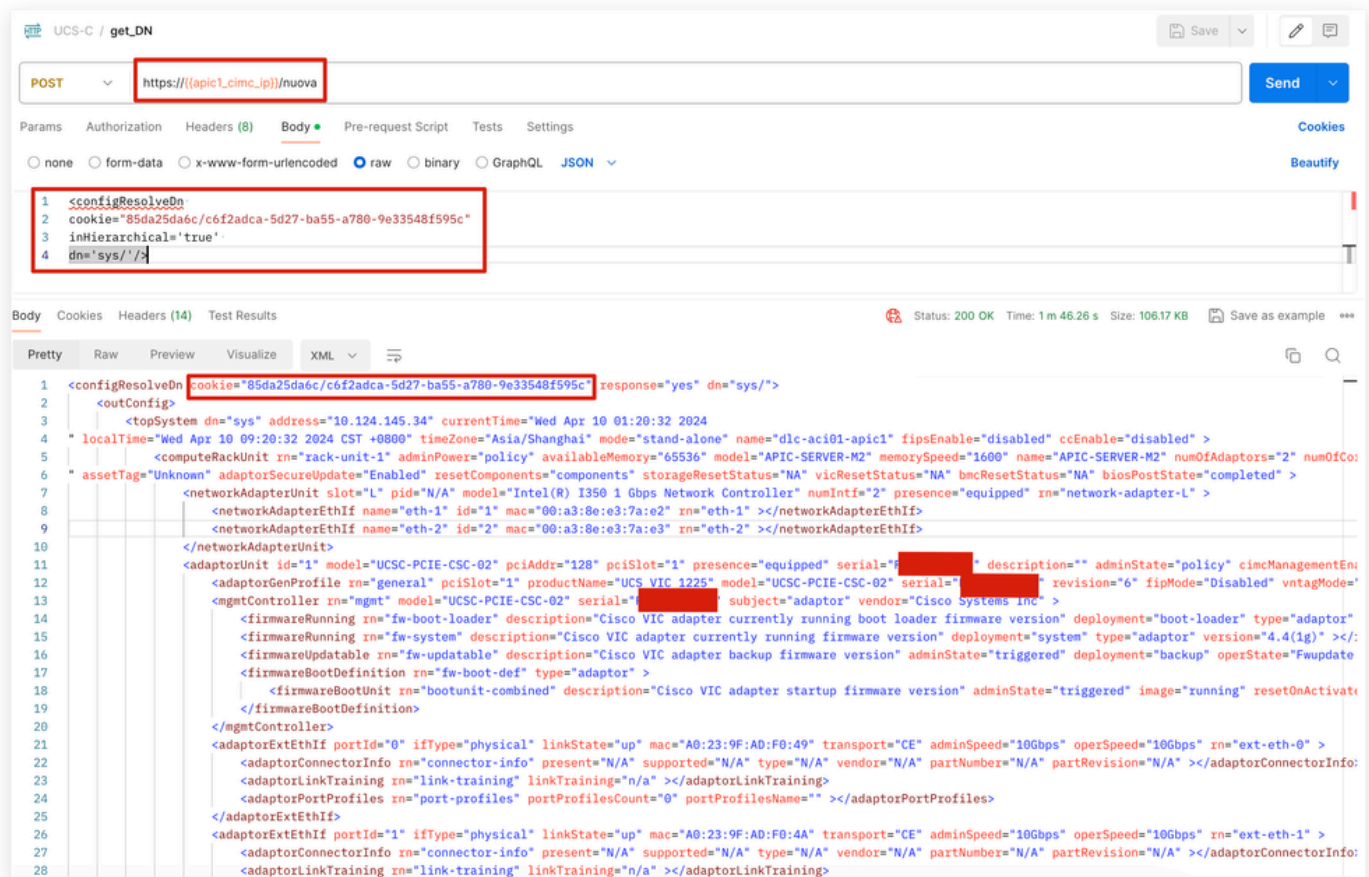
configResolveDnをinHierarchical="true"およびdn="sys/"とともに使用すると、CIMCからすべての管理オブジェクト(MO)情報が取得されます。

configResolveDn: configResolveDnメソッドは、指定されたDNの1つの管理オブジェクトを取得します。

inHierarchical=true: trueに設定すると、すべての子ノード情報が返されます。この組み合わせにより、すべてのノードおよびサブノードのMO情報をCIMCから取得できます。

dn="sys/" : これはMITの最上位ルートです。

API応答 :



Postmanの実行応答をメモ帳、PyCharm、Visual Studioコードなどのテキストエディタにコピーし、MOに基づいてクラスおよびDNを後で検索できるようにします。

例1：タイムゾーンのクラスとDNを照会する

現在のCIMC GUIで設定されているタイムゾーンは「アジア/上海」です。



手順1bでPostmanから返された結果から「Asia/Shanghai」を検索します。タイムゾーンは「Asia/Shanghai」、クラスは「topSystem」、およびDNは「sys/」です。

<#root>

<configResolveDn cookie="85da25da6c/c6f2adca-5d27-ba55-a780-9e33548f595c" response="yes" dn="sys/">
<outConfig>

```
<topSystem
```

```
dn="sys"
```

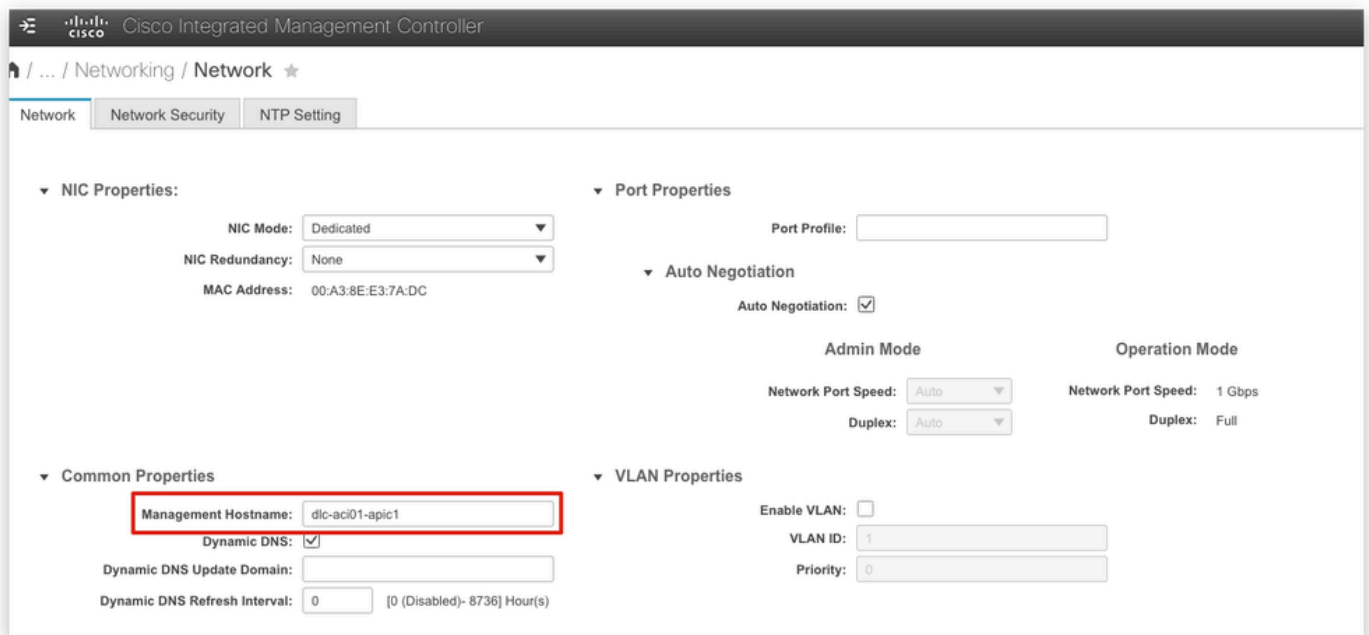
```
address="a.b.c.d" currentTime="Wed Apr 10 01:05:12 2024"  
" localTime="Wed Apr 10 09:05:12 2024 CST +0800"
```

```
timeZone="Asia/Shanghai"
```

```
mode="stand-alone" name="dlc-aci01-apic1" fipsEnable="disabled" ccEnable="disabled" >
```

例2：ホスト名のクラスとDNを照会する

現在のCIMC GUIで設定されているホスト名は「dlc-aci01-apic1」です。



Postmanから返された結果で「dlc-aci01-apic1」を検索します。ホスト名は「dlc-aci01-apic1」、クラスは「mgmtIf」、rnは「if-1」です。

```
<#root>
```

```
<mgmtIf rn="if-1" description="Management Interface Network Settings" id="1" extEnabled="yes" extIp="a.  
ifType="physical" mac="00:A3:8E:E3:7A:DC"
```

```
hostname="dlc-aci01-apic1"
```

```
dhcpEnable="no" dnsUsingDhcp="no" ddnsEnable="yes" ddnsDomain=""  
dnsPreferred="a.b.c.z" dnsAlternate="0.0.0.0" ddnsRefreshInterval="0" nicMode="dedicated" vicSlot="0" n  
vlanEnable="no" vlanId="1" vlanPriority="0" portProfile="" v6extEnabled="no" v6extIp="::" v6extGw="::"  
v6SlaacIp="::" v6dhcpEnable="no" v6dnsUsingDhcp="no" v6dnsPreferred="::" v6dnsAlternate="::" subject="b  
adminNetSpeed="auto" adminDuplex="auto" operNetSpeed="1Gbps" operDuplex="full" >
```

次に、https://CIMC_IP/visore.htmlからCIMC visoreに照会します。ホスト名「dlc-aci01-apic1」はDN='sys/rack-unit-1/mgmt/if-1'に対応します。

Filter

Class or DN: inHierarchical:

Property: Op: Val1: Val2:

[Display XML of last query](#)

Total objects shown: 1

mgmtIf ?	
dn	sys/rack-unit-1/mgmt/if-1 < >
description	Management Interface Network Settings
id	1
extEnabled	yes
extIp	
extMask	255.255.255.0
extGw	
ifType	physical
mac	00:A3:8E:E3:7A:DC
hostname	dlc-aci01-apic1
dhcpEnable	no
dnsUsingDhcp	no
ddnsEnable	yes

2. REST APIによるCIMCの管理

- ステップ1では、管理対象オブジェクト(MO)に対応するクラスと識別名(DN)がすでに特定されています。
- API経由でCIMCを管理するには、Ansible community.general.imc_restモジュールを使用できます。詳細情報は[imc_rest module ドキュメント](#)を参照してください。

APIメソッドconfigResolveClassを使用して情報を取得する

configResolveClass : メソッドは指定されたクラスの管理オブジェクトを取得します。
inHierarchical=trueの場合、結果には子が含まれます。例としてファームウェアバージョンのクエリを取り上げ、APIメソッドconfigResolveClassを使用して、MOのclassIDを指定します。

Ansibleスクリプトのコンテンツ出力 :

```
<#root>
```

```
- name: IMC login and check
community.general.imc_rest:
  hostname: '{{ imc_hostname }}'
  username: '{{ imc_username }}'
  password: '{{ imc_password }}'
  validate_certs: false # only do this when you trust the network!
```

content: |

```
<  
configResolveClass  
inHierarchical='false'  
classId='firmwareRunning'  
>
```

APIメソッドconfigConfMoを使用して設定を変更する

CIMC APIを使用してMOの設定を変更するには、configConfMoメソッドを使用します。この方法は、特定のMOの設定を構成または変更することを目的としています。configConfMoを起動する際には、変更するMOの正確なクラスまたはDN情報を指定することが重要です。

Filter

Class or DN: inHierarchical

Property: Op: Val1: Val2:

[Display XML of last query](#)

Total objects shown: 1

computeRackUnit ?	
dn	sys/rack-unit-1 < >
adminPower	policy
availableMemory	65536
model	APIC-SERVER-M2
memorySpeed	1600
name	APIC-SERVER-M2
numOfAdaptors	2
numOfCores	12
numOfCoresEnabled	12
numOfCpus	2
numOfEthHostIfs	2
numOfFcHostIfs	2
numOfThreads	12
operPower	on
originalUuid	4D1851A4-8179-418B-82B6-D1BF14460973
presence	equipped
serverId	1
serial	FCH2113V2WF
totalMemory	65536
usrLbl	ACI Lab - POD01 - dlc-aci01-apic1
uuid	4D1851A4-8179-418B-82B6-D1BF14460973

Ansibleスクリプトのコンテンツ出力 :

<#root>

```
- name: change CIMC description
community.general.imc_rest:
  hostname: '{{ imc_hostname }}'
  username: '{{ imc_username }}'
  password: '{{ imc_password }}'
  validate_certs: false
```

content: |

<

```
computeRackUnit dn="sys/rack-unit-1" usrLbl="new_lab_CIMC_description"  
/>
```

examples:

3. CIMCオートメーション設定ワークフローの例

Cisco APICは、UCS C220シリーズにインストールされるCisco ACIコントローラソフトウェアです。ワークフローは、APICソフトウェアの再イメージングの自動化プロセスを示しています。

1. Login to CIMC with pre-check
 - Retrieve firmware version
 - Retrieve faults
 - Retrieve TPM status
2. Update CIMC configurations
 - Update management hostname
 - Update Description
 - Update Timezone
 - Update ntp
 - Enable SOL
 - Update CIMC mapping vmedia
 - Update CIMC boot order to CIMC-map
 - Reboot CIMC
3. Ansible run shell expect to monitor installation status and enter iso link for APIC installation speed up
4. Retrieve CIMC post installation status
 - Update CIMC boot order back to HDD
 - Power-on host

Ansibleモジュールの例 :



注：例にはコンテンツ情報のみが含まれ、ansibleの完全なモジュールはAnsibleの公式Webサイトにあるcommunity.general.imc_restモジュールを参照します

<#root>

- name: Login to CIMC with pre-check
content: |

```
<
configResolveClass
  inHierarchical='false'
classId
='firmwareRunning'/>
```

```
<
configResolveClass
  inHierarchical='false'
classId
='faultInst'/>
```

```
<
configResolveClass
  inHierarchical='false'
classId
='equipmentTpm'/>
```

- name: IMC update CIMC infra info
content: |

<

mgmtIf

dn="sys/rack-unit-1/mgmt/if-1"

hostname="dlc-aci01-apic1"/>

<

computeRackUnit

dn="sys/rack-unit-1"

usrLbl="ACI Lab - POD01 - dlc-aci01-apic1"/>

<

topSystem

dn="sys"

timeZone="Asia/Shanghai"/>

<

commNtpProvider

dn="sys/svc-ext/ntp-svc"

ntpServer1="ntp.es1.cisco.com"/>

```
- name: Update CIMC configurations
  content: |
```

```
<
```

```
lsbootVMedia
```

```
dn="sys/rack-unit-1/boot-precision/vm-CIMC-map"
```

```
name="CIMC-map" type="VMEDIA" subtype="cimc-mapped-dvd" order="1" state="Enabled" />
```

```
<
```

```
commVMediaMap
```

```
volumeName="ACI-automation" map="www" remoteShare="http://a.b.c.d/Images/ACI/4/4.2/" remoteFile="aci-a
```

```
dn="sys/svc-ext/vmedia-svc/vmmap-ACI-automation"
```

```
>
```

```
<
```

```
computeRackUnit
```

```
dn="sys/rack-unit-1"
```

```
adminPower="hard-reset-immediate" />
```

```
# Ansible run shell expect to monitor installation status and enter iso link for APIC installation speed
```

```
- name: copy apic init script to
```

```
  template:
```

```
    src: "init.sh"
```

```
    dest: /tmp/init.sh
```

```
  delegate_to: localhost
```

```
- name: Make script executable
```

```
  file:
```

```
    path: /tmp/init.sh
```

```
    mode: "+x"
```

```
  delegate_to: localhost
```

```
  tags:
```

```
    - render
```

```
    - init
```

```
- name: Run the generated script
```

```
command: /tmp/init.sh
delegate_to: localhost
changed_when: no
tags:
  - script
```

```
- name: Retrieve CIMC post installation status
  content: |
```

<

lsbootVMedia

```
dn="sys/rack-unit-1/boot-precision/vm-CIMC-map"
name="CIMC-map" status='removed' />
```

<

commVMediaMap

```
dn="sys/svc-ext/vmedia-svc/vmmmap-ACI-automation"  
volumeName="ACI-automation" status='removed' >
```

<

lsbootStorage

```
dn="sys/rack-unit-1/boot-policy/storage-read-write"  
access="read-write" order="1" type="storage"/>
```

<

```
computeRackUnit dn="sys/rack-unit-1"  
adminPower="up" />
```

```
delegate_to: localhost  
tags:  
- retrieve_CIMC_status
```

関連情報

[Cisco UCSラックマウントサーバCisco IMC XML APIプログラマガイド](#)

[community.general.imc_restモジュール : REST APIを使用したCisco IMCハードウェアの管理](#)

[UCS Manager情報モデルリファレンス](#)

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。